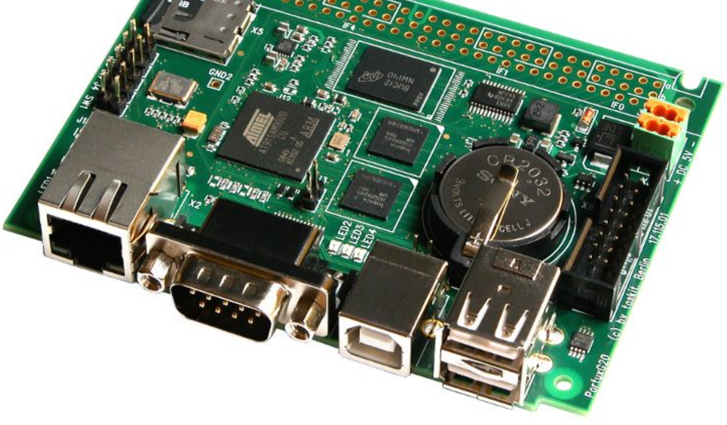


PortuxG20

Technical Reference

PortuxG20: Technical Reference

This document was written with care, but errors cannot be excluded. Neither the company named above nor the seller assumes legal liability for mistakes, resulting operational errors or the consequences thereof. Trademarks, company names and product names may be protected by law. This document may not be reproduced, edited, copied or distributed in part or in whole without written permission.

This document was generated on 2014-07-09T15:28:24+02:00.

Table of Contents

List of Figures

List of Tables

1. Introduction

The PortuxG20 is intended to be used as a universal Linux CPU card. With main peripherals like Ethernet, USB and serial port right at hand, it can be used immediately without any hardware adaptations and development. Anywhere where restricted energy and space requirements play a role, the PortuxG20 displays its strength while at the same time offering the comfort of an "easy-to-use" single board computer.

Additionally the PortuxG20 exports further interfaces to support a huge variety of peripheral devices on a 96-way bus. This allows access to custom peripherals, like 16-Bit Parallel Bus, TWI or SPI. The PortuxG20 is both flexible to custom extensions and ready-made device for immediate use.

Build in the size of half a Eurocard, it fits into 19-inch racks or other standard housings. Extension cards can be connected to the 96-way bus and sized to make up a complete Eurocard.

The ARM architecture as a modern and widely supported processor architecture is currently the platform of choice for medium performance embedded devices. Almost all major processor manufacturers have ARM products in their portfolio.

The availability of the widespread operating system "Linux" for the ARM platform opens access to a broad range of software, including tools, drivers, and software libraries. Programs written for ARM can easily be employed on the PC platform for testing and debugging. The delivered OpenEmbedded distribution offers access to a wide range of popular Linux applications.

Examples of actual or potential applications are: protocol converters, measuring and test equipment, data-logging, as well as any simple or more complex control and automation tasks.

2. Scope

This document describes the most important hardware features of the PortuxG20. It includes all informations necessary to employ customer specific hardware components for the PortuxG20. The Operating System Linux is described in a further document.

The manual comprises only a brief description of the AT91SAM9G20 processor, as this is already described in depth in the manual of the manufacturer Atmel. Descriptions of the ARM core ARM926EJ-S are available from Atmel and also at *http://www.arm.com*. It is much recommended to have a look at these documents for a thorough understanding of the processor and its integrated peripherals.

Further documentation, a developer forum and up-to-date downloads are available on our support pages at *http://www.armbedded.eu*.

3. Overview of Technical Characteristics

3.1. CPU

Atmel AT91SAM9G20 Embedded Processor featuring an ARM926EJ-S™ ARM® Thumb® Core

- CPU Clock 396 MHz
- 32KB Instruction Cache
- 32KB Data Cache
- Memory Management Unit (MMU)
- 3.3V Supply Voltage, 1.8V Memory Bus Voltage, 1.0V Core Voltage

3.2. Memory

- 128 MB NAND flash memory (optional more)
- 64 MB SDRAM (optional 128 MB)
- 2 x 16 KB SRAM
- 128 Bytes EEPROM

3.3. Interfaces and external signals

- 96-way PXB (for DIN 41612 connector)
- Ethernet 10/100 Mbit
- Dual USB 2.0 Full Speed (12 MBit/s) Hosts
- USB 2.0 Full Speed (12 MBit/s) Device
- DBGU Serial Debug Port
- 1 RS232 Serial Port
- Micro SD-Card Slot
- JTAG Debug Port
- Battery backed-up RTC

3.4. Portux Extension Bus (PXB)

- 1 Synchronous Serial Controller (SSC, I^2S)

- 1 Serial Peripheral Interface (SPI)
- 1 Two Wire Interface (TWI, I^2C)
- 1 MultiMedia Card Interfaces
- 5 USARTS
- Digital Ports - up to 64 available
- 4 Programmable Clocks
- 4-channel 10-bit ADC
- 16-Bit parallel CPU-Bus

Some of the various functions are realized by multiplexing connector pins; therefore not all functions may be used at the same time (see Appendix D, *PortuxG20 Pin Assignment*)). The 16-Bit Bus on PXB is buffered and 5V-compatible. Levels are 3.3 V.

3.5. Miscellaneous

- 4 16-Bit Timer/Counter
- Real Time Timer (RTT), with battery backup support
- Periodic Interval Timer (PIT)
- Watchdog Timer (WDT)
- Unique Hardware Serial Number
- Temperature Sensor

3.6. Power Supply

- 5V power supply
- 3V backup power supply, e.g. from a lithium battery

3.7. Dimensions

- Dimensions: 100 x 75 x 17 mm (WxDxH)

4. Hardware Description

4.1. AT91SAM9G20 Processor Core

The AT91SAM9G20 runs at 396 MHz with a memory bus frequency of 132 MHz.

Here are some of the most important features of the AT91SAM9G20's ARM926EJ-S core:

- 32 Kbyte Data Cache, 32 Kbyte Instruction Cache, Write Buffer
- 32 Bit Data Bus
- ARM v4 and v5 Memory Management Unit (MMU)
- ARM v5 32-bit Instruction Set, ARM Thumb 16-bit Instruction Set supported
- DSP Instruction Extensions
- ARM Jazelle® Technology for Java® Acceleration
- EmbeddedICE™ Debug Communication Channel Support

Some of these features - like Jazelle - are currently not supported by the operating system of the product.

4.2. Memory

The PortuxG20 is equipped with 32-Bit CPU-bus. Only a 16-Bit bus is exported on the interface connectors of the PortuxG20. The memory bus voltage is 1.8 V and runs at 132 MHz. The memory bus voltage is different from normal operating voltage, which is 3.3 V. This has to be considered, when designing additional peripherals connected to the memory bus. Eventually buffer chips are necessary.

4.2.1. NAND Flash

The PortuxG20 is equipped with a 128 MB NAND flash with 100000 erase and write cycles. It is organized in 128KB blocks. Customer specific adaptations are possible up to 512 MB on-board NAND flash. It is connected to chip select three (NCS3) of the micro-controller.

NAND flash has a different organisation of transistors than the commonly used NOR flash. While it allows a much higher density and thus an increase in storage capacity, there are some differences which need to be kept in mind.

Typically, NAND flash is organized in pages and blocks, similar to hard disks. Pages are 512, 2048 or 4096 bytes in size, typical block sizes are 16, 128, 256 or 512 KB. Reading and programming are performed on a page basis. Programming can only be done sequently in one block.

Additionally, NAND flash requires bad block management, either by the driver software or by a separate controller chip. Most NAND devices are shipped with bad blocks. These are identified and marked according to a specified bad block strategy. Further bad blocks may be detected during runtime. They are detected via an ECC (error correcting code). If a bad block is detected, the data is written to a different, good block, and the bad block table is updated. So the overall memory capacity gradually shrinks as more and more blocks are marked bad.

This error detection is done by software like U-boot and Linux. Additionally, NAND flash is subject to a limited number of write and erase cycles. These are typically 100.000 cycles per block. So it is highly recomended to use wear levelling filesystems.

4.2.2. SDRAM

The PortuxG20 is equipped with 64MB SDRAM. Customer specific adaptations allow configurations up to 128MB. The SDRAM is connected to chip select one (NCS1) of the micro-controller.

SDRAM is volatile memory which allows random access to any location of its memory area. SDRAM has a synchronous interface. This means that it waits for a clock signal before responding to its control inputs, therefore it is synchronized with the CPU bus. The clock is used to drive a finite state machine in the chip, which allows to accept new instructions, before the previous one has finished executing.

4.2.3. EEPROM

The PortuxG20 is equipped with a 128 bytes EEPROM, connected to the Dallas™ 1 wire bus.

EEPROM stands for Electrically Erasable Programmable Read-Only Memory and is non-volatile memory, which is used to store small amounts of data like calibration or configuration data. EEPROMS are byte-wise erasable, thus allowing true random access.

4.2.4. SRAM

The PortuxG20's micro-controller is equipped with 2 x 16 KB internal SRAM. The internal SRAM can be accessed in one bus cycle and may be used for time critical sections of code or interrupt handlers.

4.3. Bus Matrix

The bus matrix of AT91SAM-controllers allows many master and slave devices to be connected independently of each other. Each master has a decoder and can be defined specially for each master. This allows concurrent access of masters to their slaves (provided the slave is available).

The bus matrix is thus the bridge between external devices connected to the EBI, the microcontroller's embedded peripherals and the CPU core.

Master 0	ARM926™ Instruction
Master 1	ARM926™ Data

Master 2	PDC
Master 3	ISI Controller
Master 4	Ethernet MAC
Master 5	USB Host DMA

Table 4.1. Bus Matrix Masters

Slave 0	Internal SRAM0 16KB
Slave 1	Internal SRAM1 16KB
Slave 2	Internal ROM / USB Host User Interface
Slave 3	External Bus Interface (EBI)
Slave 4	Internal Peripherals

Table 4.2. Bus Matrix Slaves

4.4. Advanced Interrupt Controller (AIC)

The core features of the Advanced Interrupt Controller are:

- 32 Internal or External Interrupt Sources
- 8-level Priority Controller
- Level Sensitive or Edge Triggered
- Programmable Polarity for External Sources

Moreover, all PIO lines can be used to generate a PIO interrupt. However, the PIO lines can only generate level change interrupts, that is, positive as well as negative edges will generate an interrupt. The PIO interrupt itself (PIO to AIC line) is usually programmed to be level-sensitive. Otherwise interrupts will be lost if multiple PIO lines source an interrupt simultaneously.

On the PortuxG20 IRQ0, IRQ2 and the FIQ are available. The list of peripheral identifiers, which are used to program the AIC can be found in Table B.1, “Peripheral Identifiers”

4.5. Battery Backup

The following parts of the AT91SAM9G20 Processor can be backed-up by a battery:

- Slow Clock Oscillator
- Real Time Timer
- Reset Controller
- Shutdown Controller
- General Purpose Backup Registers

It is recommended to always use a backup power supply (normally a battery) in order to speed up the boot-up time and to avoid reset problems.

4.6. Reset Controller (RSTC)

The embedded microcontroller has an integrated Reset Controller which samples the backup and the core voltage. The presence of a backup voltage (VDDBU) when the card is powered down speeds up the boot time of the microcontroller.

4.7. Serial Number

Every PortuxG20 has a unique 48-bit hardware serial number chip which can be used by application software. The chip is a Dallas® one-wire-chip. A Linux driver is provided. Additional it functions as the 128 Byte EEPROM.

4.8. Clock Generation

4.8.1. Processor Clocks

The CPU generates its clock signals based on two crystal oscillators: One slow clock (SLCK) oscillator running at 32.768 KHz and one main clock oscillator running at 18.432 MHz. The slow clock oscillator also serves as the time base for the real time timer. It draws a minimum of current (a few micro-Amps) and can therefore be backeded up by a small lithium battery when the board is powererd down.

From the main clock oscillator, the CPU generates two further clocks by using two PLLs. PLLA provides the processor clock (PCK) and the master clock (MCK). PLLB typically provides the 48 MHz USB clock and is normally used only for this purpose. The clocks of most peripherals are derived from MCK. These include EBI, USART, SPI, TWI, SSC, PIT and TC.

Some peripherals like the programmable clocks and the timer counters (TC) can also run on SLCK. The real time timer (RTT) always runs on SLCK.

Clock	Frequency	Source
PCK (Processor Clock)	396 MHz	PLLA
MCK (Master Clock)	132 MHz	PCK/3
USB Clock	48 MHz	PLLB
Slow Clock	32.768 KHz	Slow Clock Oscillator

Table 4.3. AT91SAM9G20 Clocks

4.8.2. Programmable Clocks

The programmable clocks can be individually programmed to derive their input from SLCK, PLLA, PLLB and Main Clock. Each PCK has a divider of 2, 4, 8, 16, 32 or 64.

The PortuxG20 features two programmable clocks PCK0, PCK1.

4.9. Power Management Controller (PMC)

4.9.1. Function

The PMC has a Peripheral Clock register which allows to individually enable or disable the clocks of all integrated peripherals by using their "Peripheral Identifier" (see Table B.1, “Peripheral Identifiers”). The System Clock register allows to enable or disable each of the following clocks individually:

- Processor Clock
- ISI Clock
- USB Host Clock (common for both channels)
- USB Device Clock
- Programmable Clocks

The PMC status register provides "Clock Ready" or, respectively, "PLL Lock" status bits for each of these clocks. An interrupt is generated when any of these bits changes from 0 to 1. The PMC provides status flags for the

- Main Oscillator
- Master Clock
- PLLA
- PLLB
- Programmable Clocks

The Main Oscillator frequency can be measured by using the PMC Main Clock Frequency register. The SLCK is used as reference for the measurement.

4.9.2. Power Management

Using power management can dramatically reduce the power consumption of an Embedded Device. Via the PMC various clocks can be disabled or their speed can be reduced:

- stopping the PLLs (PLLA and / or PLLB)
- stopping the clocks of the various peripherals
- reducing the clock rates of peripherals, especially by changing MCK.

The PMC supports the following power-saving features: Idle mode and power-down mode. Please note that not every operating system supports these modes.

- **Idle Mode.** In idle mode, the processor clock will be re-enabled by any interrupt. The peripherals, however, are only able to generate an interrupt if they still have a clock, so care has to be taken as to when a peripheral can be powered down.

- **Power-down Mode.** In many cases a system waits for a user action or some other rare event. In such a case, it is possible to change MCK to SLCK. Any external event which changes the state on peripheral pins (not the USB) can then be detected by the PIO controller or the AIC.

  It should also be taken into account that when a PLL is stopped it will take some time to restart it. Changing the PLL frequencies or stopping them can therefore be done only at a moderate rate. If short reaction times are required, this is not a choice.

Additionally, the following measures can reduce power consumption considerably:

- switching off the TFT supply voltage
- putting peripheral chips like Ethernet controller and / or PHY or serial driver devices in power down mode
- putting the SDRAM into self-refresh mode

4.10. Real-time Timer (RTT)

The Real-time Timer is a 32-bit counter combined with a 16-bit prescaler running at Slow Clock (SLCK = 32768 Hz). As the RTT keeps running if only the backup supply voltage is available, it is used as a Real-time clock.

The RTT can generate an interrupt every time the prescaler rolls over. Usually the RTT is configured to generate an interrupt every second, so the prescaler will be programmed with the value 7FFFh.

The RTT can also generate an alarm if a preprogrammed 32-bit value is reached by the counter.

4.11. Timer Counter (TC)

The PortuxG20 features two blocks of timer counters with three counters each. Due to multiplexing four timer counters may be used with external signals.

The TC consists of three independent 16-bit Timer/Counter units. They may be cascaded to form a 32-bit or 48-bit timer/counter. The timers can run on the internal clock sources MCK/2, MCK/8, MCK/32, MCK/128, SLCK or the output of another timer channel. External clocks may be used as well as the counters can generate signals on timer events. They also can be used to generate PWM signals.

4.12. Periodic Interval Timer (PIT)

The PIT consists of a 20-bit counter running on MCK / 16. This counter can be preloaded with any value between 1 and 2^{20}. The counter increments until the preloaded value is reached. At this stage it rolls over and generates an interrupt. An additional 12-bit counter counts the interrupts of the 20 bit counter.

The PIT is intended for use as the operating system’s scheduler interrupt.

4.13. Watchdog Timer

The watchdog timer is a 12-bit timer running at 256 Hz (Slow Clock / 128). The maximum watchdog timeout period is therefore equal to 16 seconds. If enabled, the watchdog timer asserts a hardware reset at the end of the timeout period. The application program must always reset the watchdog timer before the timeout is reached. If an application program has crashed for some reason, the watchdog timer will reset the system, thereby reproducing a well defined state once again.

The Watchdog Mode Register can be written only once. After a processor reset, the watchdog is already activated and running with the maximum timeout period. Once the watchdog has been reconfigured or deactivated by writing to the Watchdog Mode Register, only a processor reset can change its mode once again.

4.14. Peripheral DMA Controller (PDC)

The Peripheral DMA Controller (PDC) transfers data between on-chip serial peripherals and the on- and/or off-chip memories. The PDC contains unidirectional and bidirectional channels. The full-duplex peripherals feature unidirectional channels used in pairs (transmit only or receive only). The half-duplex peripherals feature one bidirectional channel. Typically full-duplex peripherals are USARTs, SPI or SSC. The MCI is a half duplex device.

The user interface of each PDC channel is integrated into the user interface of the peripheral it serves. The user interface of unidirectional channels (receive only or transmit only), contains two 32-bit memory pointers and two 16-bit counters, one set (pointer, counter) for current transfer and one set (pointer, counter) for next transfer. The bidirectional channel user interface contains four 32-bit memory pointers and four 16-bit counters. Each set (pointer, counter) is used by current transmit, next transmit, current receive and next receive.

Using the PDC removes processor overhead by reducing its intervention during the transfer. This significantly reduces the number of clock cycles required for a data transfer, which improves microcontroller performance. To launch a transfer, the peripheral triggers its associated PDC channels by using transmit and receive signals. When the programmed data is transferred, an end of transfer interrupt is generated by the peripheral itself. There are four kinds of interrupts generated by the PDC:

- End of Receive Buffer
- End of Transmit Buffer
- Receive Buffer Full
- Transmit Buffer Empty

The "End of Receive Buffer" / "End of Transmit Buffer" interrupts signify that the DMA counter has reached zero. The DMA pointer and counter register will be reloaded from the reload registers ("DMA new pointer register" and "DMA new counter register") provided that the "DMA new counter register" has a non-zero value. Otherwise a "Receive Buffer Full" or, respectively, a "Transmit Buffer Empty" interrupt is generated, and the DMA transfer terminates. Both reload registers are set to zero automatically after having been copied to the DMA pointer and counter registers.

4.15. Debug Unit (DBGU)

The Debug Unit is a simple UART which provides only RX/TX lines. It is used as a simple serial console for Firmware and Operating Systems.

4.16. JTAG Unit

The JTAG unit can be used for hardware diagnostics, hardware initialization, flash memory programming, and debug purposes. The JTAG unit supports two different modes, namely the "ICE Mode", and the "Boundary Scan" mode. It is normally jumpered for "ICE Mode".

JTAG interface devices are available for the unit. However, the use of them is not within the scope of this document.

4.17. Two-wire Interface (TWI)

The TWI is also known under the expression "I^2C-Bus", which is a trademark of Philips and may therefore not be used by other manufacturers. However, interoperability is guaranteed. The TWI supports both master or slave mode.

The TWI uses only two lines, namely serial data (SDA) and serial clock (SCL). According to the standard, the TWI clock rate is limited to 400 kHz in fast mode and 100 kHz in normal mode, but configurable baud rate generator permits the output data rate to be adapted to a wide range of core clock frequencies.

4.18. Multimedia Card Interface (MCI)

The PortuxG20 features a on-board Micro-SD-Card slot, which is connected to the MCI-A interface of the micro-controller. Also operating systems like Linux do not necessarily support all features of the hardware unit.

The MultiMedia Card Interface (MCI) supports the MultiMedia Card (MMC) Specification V3.11, the SDIO Specification V1.1 and the SD Memory Card Specification V1.0.

The MCI includes a command register, response registers, data registers, timeout counters and error detection logic that automatically handle the transmission of commands and, when required, the reception of the associated responses and data with a limited processor overhead. The MCI supports stream, block and multi-block data read and write, and is compatible with the Peripheral DMA Controller (PDC) channels, minimizing processor intervention for large buffer transfers.

The MCI operates at a rate of up to Master Clock divided by 2 and supports the interfacing of 2 slot(s). Each slot may be used to interface with a MultiMediaCard bus (up to 30 Cards) or with a SD Memory Card. Only one slot can be selected at a time (slots are multiplexed). A bit field in the SD Card Register performs this selection.

The SD Memory Card communication is based on a 9-pin interface (clock, command, four data and three power lines) and the MultiMedia Card on a 7-pin interface (clock, command, one data, three power lines and one reserved for future use). The SD Memory Card interface also supports MultiMedia Card operations. The main differences between SD and MultiMedia Cards are the initialization process and the bus topology.

4.19. USB Host Port (UHP)

The PortuxG20 integrates two USB host ports supporting speeds up to 12 MBit/s.

The USB Host Port (UHP) interfaces the USB with the host application. It handles Open HCI protocol (Open Host Controller Interface) as well as USB v2.0 Full-speed and Low-speed protocols.

The USB Host Port integrates a root hub and transceivers on downstream ports. It provides several high-speed half-duplex serial communication ports. Up to 127 USB devices (printer, camera, mouse, keyboard, disk, etc.) and an USB hub can be connected to the USB host in the USB "tiered star" topology.

4.20. USB Device Port (UDP)

The PortuxG20 integrates one USB device port supporting speeds up to 12 MBit/s.

The USB Device Port (UDP) is compliant with the Universal Serial Bus (USB) V2.0 full-speed device specification. The USB device port enables the product to act as a device to other host controllers.

The USB device port can also be implemented to power on the board. One I/O line may be used by the application to check that VBUS is still available from the host. Self-powered devices may use this entry to be notified that the host has been powered off. In this case, the pullup on DP must be disabled in order to prevent feeding current to the host. The application should disconnect the transceiver, then remove the pullup.

4.21. Ethernet MAC (EMAC)

The EMAC module implements a 10/100 MBit/s Ethernet MAC compatible with the IEEE 802.3 standard using an address checker, statistics and control registers, receive and transmit blocks, and a DMA interface.

The address checker recognizes four specific 48-bit addresses and contains a 64-bit hash register for matching multicast and unicast addresses. It can recognize the broadcast address of all ones, copy all frames, and act on an external address match signal.

An individual 48-bit MAC address (ETHERNET hardware address) is allocated to each product. This number is stored in flash memory. It is recommended not to change the MAC address in order to comply with IEEE Ethernet standards.

4.22. Universal Sychronous Asynchronous Receiver and Transmitter (USART)

The PortuxG20 has up to five independent USARTs, not including the debug unit. One USART is available in RS232 levels.

The Universal Synchronous Asynchronous Receiver Transceiver (USART) provides one full duplex universal synchronous asynchronous serial link. Data frame format is widely programmable (data length, parity, number of stop bits) to support a maximum of standards. The receiver implements parity error, framing error and overrun error detection. The receiver time-out enables handling variable-length frames and the transmitter timeguard facilitates communications with slow remote devices. Multidrop communications are also supported through address bit handling in reception and transmission.

The USART supports the connection to the Peripheral DMA Controller, which enables data transfers to the transmitter and from the receiver. The PDC provides chained buffer management without any intervention of the processor.

Six different modes are implemented within the USARTs:

- Normal (standard RS232 mode)
- RS485
- Hardware Handshaking
- ISO7816 Protocol: T=0 or T=1
- IrDA

RS485. In RS485 operating mode the RTS pin is automatically driven high during transmit operations. If RTS is connected to the "enable" line of the RS485 driver, the driver will thus be enabled only during transmit operations.

Hardware Handshaking. The hardware handshaking feature enables an out-of-band flow control by automatic management of the pins RTS and CTS. The receive DMA channel must be active for this mode. The RTS signal is driven high if the receiver is disabled or if the DMA indicates a buffer full condition. As the RTS signal is connected to the CTS line of the connected device, its transmitter is thus prevented from sending any more characters.

ISO7816. The USARTs have an ISO7816-compatible mode which permits interfacing with smart cards and Security Access Modules (SAM). Both T=0 and T=1 protocols of the ISO7816 specification are supported.

IrDA. The USART features an infrared (IrDA) mode supplying half-duplex point-to-point wireless communication. It includes the modulator and demodulator which allows a glueless connection to the infrared transceivers. The modulator and demodulator are compliant with the IrDA specification version 1.1 and support data transfer speeds ranging from 2.4 kb/s to 115.2 kb/s.

Signals of the Serial Interfaces. All UARTs/USARTs have one receiver and one transmitter data line (full duplex). Not all USARTs are implemented with full modem control lines. Furthermore the available lines depend largely on the used multiplexing. Most modem control lines can be implemented with standard digital ports.

Hardware Interrupts. There are several interrupt sources for each USART:

- Receive: RX Ready, (DMA) Buffer Full, End of Receive Buffer

- Transmit: TX Ready, (DMA) Buffer Empty, End of Transmit Buffer, Shift Register Empty
- Errors: overrun, parity, framing, and timeout errors
- Handshake: the status of CTS has changed
- Break: the receiver has detected a break condition on RXD
- NACK: non acknowledge (ISO7816 mode only)
- Iteration: the maximum number of repetitions has been reached (ISO7816 mode only)

Please refer to the chapter about the DMA unit (PDC) for a description of the "Buffer Full" and "End of Receive / Transmit Buffer" events.

4.23. Synchronous Peripheral Interface (SPI)

The PortuxG20 features one SPI port with three chip selects.

The Serial Peripheral Interface (SPI) circuit is a synchronous serial data link that provides communication with external devices in Master or Slave Mode. It also enables communication between processors if an external processor is connected to the system.

The Serial Peripheral Interface is essentially a shift register that serially transmits data bits to other SPIs. During a data transfer, one SPI system acts as the "master" which controls the data flow, while the other devices act as "slaves" which have data shifted into and out by the master.

A slave device is selected when the master asserts its NSS signal. If multiple slave devices exist, the master generates a separate slave select signal for each slave (NPCS). The SPI system consists of two data lines and two control lines:

- Master Out Slave In (MOSI): This data line supplies the output data from the master shifted into the input(s) of the slave(s).
- Master In Slave Out (MISO): This data line supplies the output data from a slave to the input of the master. There may be no more than one slave transmitting data during any particular transfer.
- Serial Clock (SPCK): This control line is driven by the master and regulates the flow of the data bits. The master may transmit data at a variety of baud rates; the SPCK line cycles once for each bit that is transmitted. The SPI baudrate is Master Clock (MCK) divided by a value between 1 and 255
- Slave Select (NSS): This control line allows slaves to be turned on and off by hardware.

Each SPI Controller has a dedicated receive and transmit DMA channel.

4.24. Synchronous Serial Controller (SSC)

The PortuxG20 has one SSC interface available, depending on the multiplexing of the pins.

The SSC supports many serial synchronous communication protocols generally used in audio and telecom applications such as I2S, Short Frame Sync, Long Frame Sync, etc.

The SSC has separated receive and transmit channels. Each channel has a data, a clock and a frame synchronization signal (RD, RK, RF, resp. TD, TK, TF). Both a receive and a transmit DMA channel are assigned to each SSC.

4.25. Peripheral Input/Output Controller (PIO)

The PortuxG20 has a maximum of 64 freely programmable digital I/O ports on its connectors. These pins are also used by other peripheral devices.

The Parallel Input/Output Controller(PIO) manages up to 32 programmable I/O ports. Each I/O port is associated with a bit number in the 32 bit register of the user interface. Each I/O port may be configured for general purpose I/O or assigned to a function of an integrated peripheral device. In doing so multiplexing with multiple integrated devices is possible. That means a pin may be used as GPIO or only as one of the peripheral functions. The PIO Controller also features a synchronous output providing up to 32 bits of data output in a single write operation.

The following characteristics are individually configurable for each PIO pin:

- PIO enable
- Peripheral enable
- Output enable
- Output level
- Write Enable
- Level change interrupt
- Glitch filter: pulses that are lower than a half clock cycle are ignored
- Open-drain outputs
- Pull-up resistor

All configurations as well as the pin status can be read back by using the appropriate status register. Multiple pins of each PIO can also be written simultaneously by using the synchronous output register.

For interrupt handling, the PIO Controllers are considered as user peripherals. This means that the PIO Controller interrupt lines are connected among the interrupt sources 2 to 31. Refer to the PIO Controller peripheral identifier Table B.1, “Peripheral Identifiers” to identify the interrupt sources dedicated to the PIO Controllers. The PIO Controller interrupt can be generated only if the PIO Controller clock is enabled.

A number of the PIO signals might be used internally on the module. Care has to be taken when accessing the PIO registers in order not to change the settings of these internal signals, otherwise a system crash is likely to happen.

4.26. Analog Digital Converter (ADC)

The PortuxG20 has a four ADC channels available. On PortuxG20 ADVREF is 3.3V. The trigger ADTRG is not available.

The ADC is based on a Successive Approximation Register (SAR) 10-bit Analog-to-Digital Converter(ADC). It also integrates a 4-to-1 analog multiplexer, making possible the analog-to-digital conversions of 4 analog lines. The conversions extend from 0V to ADVREF.

The ADC supports an 8-bit or 10-bit resolution mode, and conversion results are reported in a common register for all channels, as well as in a channel-dedicated register. Software trigger, external trigger on rising edge of the ADTRG pin or internal triggers from Timer Counter output(s) are configurable.

The ADC also integrates a Sleep Mode and a conversion sequencer and connects with a DMA channel. These features reduce both power consumption and processor intervention.

Finally, the user can configure ADC timings, such as Startup Time and Sample and Hold Time.

5. Layout Description

5.1. Layout Diagram

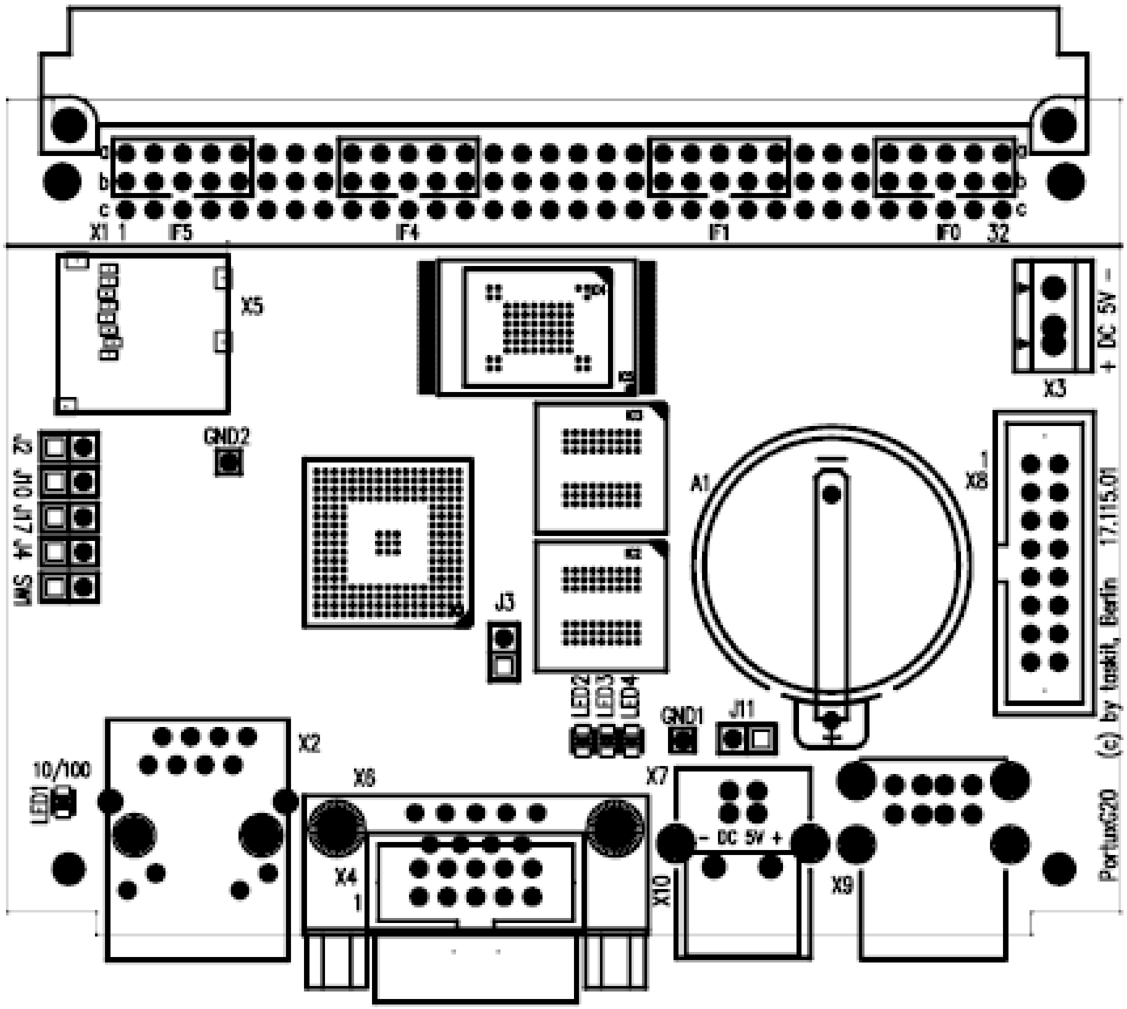


Figure 5.1. PortuxG20 Layout Diagram

5.2. PortuxG20 Connector Description

5.2.1. Pin-header X1

Pin-header X1 has 96 contacts in 2.54 mm (0,1 inch) grid pitch. It can be used to assemble DIN 41612 standard (IEC/EN 60603-2) connectors. The grid is organized in three rows (row a, b and c) with 32 contacts each. The pin assignment of this connector is described in Appendix D, *PortuxG20 Pin Assignment*.

In the default state no connector is assembled at all on the PortuxG20 to allow flexible use of the pins for the customer. The AT91SAM9G20's USARTs are organized in four ten-way connectors which can be assembled alternatively. They are named IF0, IF1, IF4 and

IF5 and correspond to the AT91SAM9G20's USART0, 1,4 and 5 respectively. Samples for both connectors are contained in the Starterkit.

All levels of USARTs on X1 are low-level TTL (3.3V). In the following pin assignments, pins driven by the hardware IP of the AT91SAM9G20 are coloured. If further signals are needed they must be programmed individually by the user.

5.2.1.1. IF Default Assignment

Pin	Assignment
1	VCC
2	-DSR
3	RI
4	RXD
5	TXD
6	-DTR
7	-RTS
8	-CTS
9	-DCD
10	GND

Table 5.1. IF Pin Assignment

5.2.1.2. IF0 Pin Assignment

Pin	Assignment	X1-Pin	Processor Pin
1	VCC	B28	VCC
2	-DSR	A28	PB22/DSR0/ISI_D2
3	RI	B29	PB25/RI0/ISI_D5
4	RXD	A29	PB05/RXD0
5	TXD	B30	PB04/TXD0
6	-DTR	A30	PB24/DTR0/ISI_D4
7	-RTS	B31	PB26/RTS0/ISI_D6
8	-CTS	A31	PB27/CTS0/ISI_D7
9	-DCD	B32	PB23/DCD0/ISI_D3
10	GND	A32	GND

Table 5.2. IF0 Pin Assignment

5.2.1.3. IF1 Pin Assignment

Pin	Assignment	X1-Pin	Processor Pin
1	VCC	B20	VCC
2	-DSR	A20	PB17/TF0/TCLK4
3	RI	B21	PB16/TK0/TCLK3
4	RXD	A21	PB07/RXD1/TCLK2
5	TXD	B22	PB06/TXD1/TCLK1

Pin	Assignment	X1-Pin	Processor Pin
6	-DTR	A22	PB19/RD0/TIOB5
7	-RTS	B23	PB06/RTS1/ISI_PCK
8	-CTS	A23	PB29/CTS1/ ISI_VSYNC
9	-DCD	B24	PB18/TD0/TIOB4
10	GND	A24	GND

Table 5.3. IF1 Pin Assignment

5.2.1.4. IF4 Pin Assignment

Pin	Assignment	X1-Pin	Processor Pin
1	VCC	B9	VCC
2	-DSR	A9	PB21/RF0/ISI_D1
3	RI	B10	PB20/RK0/ISI_D0
4	RXD	A10	PA30/SCK2/RXD4
5	TXD	B11	PA31/SCK0/TXD4
6	-DTR	A11	PA03/SPI0_NPCS0/ MCDB3
7	-RTS	B12	PA02/SPI0_SPCK
8	-CTS	A12	PA01/SPI0_MOSI/ MCCDB
9	-DCD	B13	PA00/SPI0_MISO/ MCDB0
10	GND	A13	GND

Table 5.4. IF4 Pin Assignment

5.2.1.5. IF5 Pin Assignment

Pin	Assignment	X1-Pin	Processor Pin
1	VCC	B1	VCC
2	-DSR	A1	PB31/PCK1/ISI_MCK
3	RI	B2	PB30/PCK0/ ISI_HYSNC
4	RXD	A2	PB13/RXD5/ISI_D11
5	TXD	B3	PB12/TXD5/ISI_D10
6	-DTR	A3	PB03/ SPI1_NPCS0/TIOA5
7	-RTS	B4	PB02/ SPI1_SPCK/TIOA4
8	-CTS	A4	PB01/ SPI1_MOSI/TIOB3
9	-DCD	B5	PB00/ SPI1_MISO/TIOA3
10	GND	A5	GND

Table 5.5. IF Pin Assignment

5.2.2. X2 RJ45 Jack

RJ45 jack for 10/100 MBit/s ethernet.

5.2.3. X3 DC 5V

Mini spring-clamp connection for 5V regulated DC-IN.

Warning

Do not connect voltages above the specified 5V to the board. Overvoltage can result in damages beyond repair.

5.2.4. Pin-header X4

Pin-header X4 is an alternative assembling for the DSUB serial connector X6. Levels are RS232.

Pin	Assignment	Processor Pin
1	-DCD	PA22/ADTRG/ETXFR
2	DRXD	PB14/DRXD
3	RXD	PB09/RXD2
4	-RTS	PA04/RTS2/MCDB2
5	TXD	PB08/TXD2
6	-CTS	PA05/CTS2/MCDB1
7	DTXD	PB15/DTXD
8	-RI	PA25/TCLK0/ERX2
9	GND	GND
10	VNC	VCC

Table 5.6. X4 Pin Assignment

5.2.5. X5 Micro SD-Card Slot

Four wire Micro SD-Card slot.

5.2.6. Pin-header X6

Pin-header X6 is a DSUB-connector in RS232 levels. It corresponds to USART2 of the AT91SAM9G20. On the original signals DSR and DTR is the AT91SAM9G20's debug serial port (DBGU). An adapter is needed (DBGU-Adapter) to connect to the DBGU. The adapter is part of the Starterkit contents. Schematics to build one can be found in Appendix E, *Schematics DBGU Adaptor*. With Jumper 6 (not assembled by default) RI can be powered by VCC 3.3V.

Pin	Assignment	Processor Pin
1	-DCD	PA22/ADTRG/ETXFR
2	RXD	PB09/RXD2
3	TXD	PB08/TXD2

Pin	Assignment	Processor Pin
4	DTXD	PB15/DTXD
5	GND	GND
6	DRXD	PB14/DRXD
7	-RTS	PA04/RTS2/MCDB2
8	-CTS	PA05/CTS2/MCDB1
9	-RI	PA25/TCLK0/ERX2
10	SERGND	
11	SERGND	

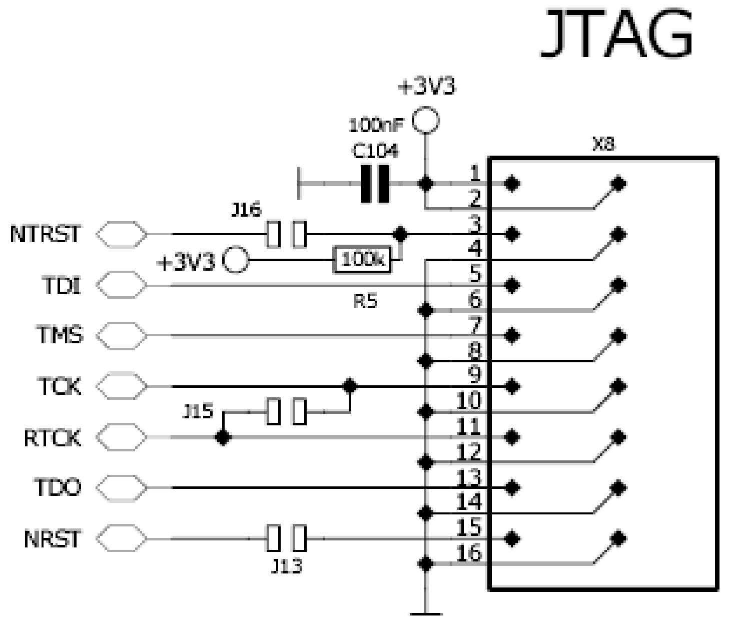
Table 5.7. X6 Pin Assignment

5.2.7. X7 USB Device Port

Full speed USB device port.

5.2.8. Pin-header X8 JTAG

16-way connector in 2.54 mm pitch grid. It allows access to the JTAG-Unit of the AT91SAM9G20. J16, J15 and J13 are not assembled.



5.2.9. X9 Dual USB Host Ports

Two full speed USB host ports.

5.2.10. A1 Battery Holder

Battery holder for CR 2032 battery for RTC backup.

5.2.11. GND1 and GND2

GND test pins

5.2.12. LEDs

LED	Role	Assignment
LED1	ETH Speed Status	
LED2	LED red	BPC05/A24/SPI1_NPCS1
LED3	LED green	BPC04/A23/SPI1_NPCS2
LED4	LED yellow	BPC10/A25/CTS3

Table 5.8. LED

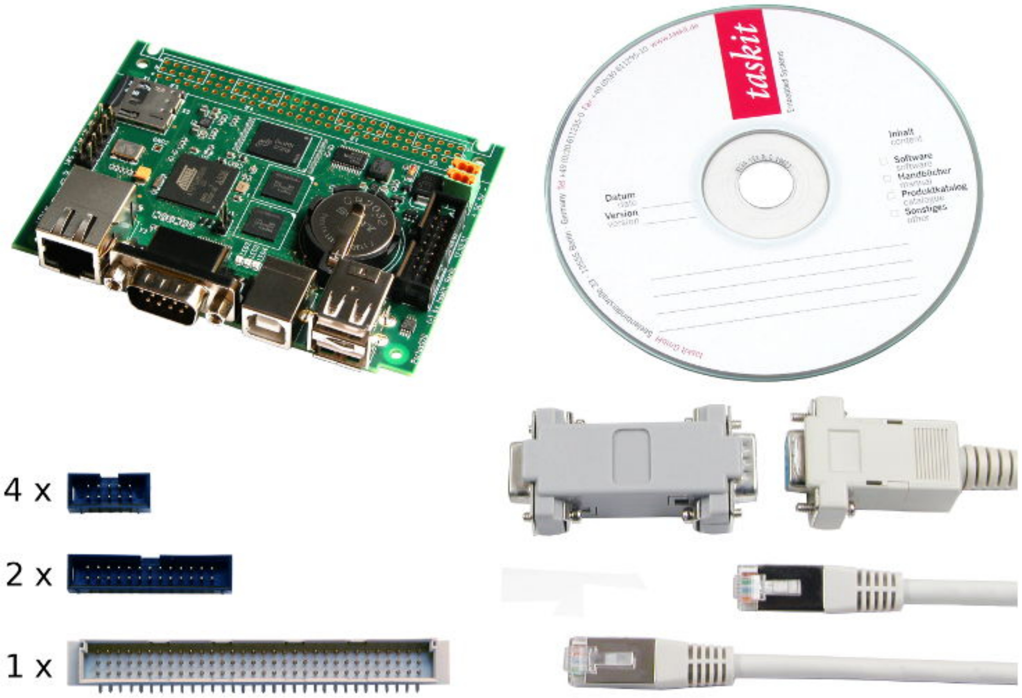
5.2.13. Jumper

Jumper	Role	Assignment	Default
J2	NRST Low	NRST	Unset
J3	Disable NCS3	**PC14/NCS3/IRQ2**	Unset
J4	WKUP Low	WKUP	Unset
J10	BMS Low	BMS	Unset
J11	Power over USBD		Set
J17	SHDN Low	SHDN	Unset
SW1	GPIO Switch	BPC06/TIOB2/CFCE1	Unset

Table 5.9. Jumper

6. PortuxG20 Starterkit

6.1. Starterkit Contents



The PortuxG20 Starterkit contains the following components:

- PortuxG20
- 2GB Micro SD card with root file system
- Four 10-way straight 2.54mm pitch grid connectors
- Two 26-way straight 2.54mm pitch grid connectors
- 96-way angled EN 60603-2 connector
- Wall Adapter Power Supply for USB Device Port, Input AC 230V, Output 5V, min. 500 mA
- Ethernet patch cable (cross and straight)
- Serial "Null-Modem" Cable with two 9-pin D-type connectors
- DBGU-Adaptor for accessing the Debug UART
- CD with Operating System, Toolchain, and Documentation

Appendix A. Peripheral Color Codes

This table matches the color used to identify various peripherals in tables.

Power Supply/Ground
USART
Debug UART
TWI (I^{2}C-Bus)
SD-Card/MMC
SPI
USB Host
USB Device
Reserved
Synhcronous Serial Controller (SSC)
JTAG
Control
Ethernet
Genral Purpose I/O Port
Programmable Clock Output
Analog-to-digital Converter
Timer Counter
Image Sensor Interface
LCD/TFT Controller Interface
Embedded Trace Macrocell
Static Memory Controller
Compact Flash Interface
Pulse Width Modulator
Touch Controller
Can Controller
AC97 Sound Interface
Encryption Device
Soft Modem
True Random Generator

Appendix B. Peripheral Identifiers

ID	Mnemonic	Peripheral Name	External Interrupt
0	AIC	Advanced Interrupt Controller	FIQ
1	SYSC	System Controller Interrupt	
2	PIOA	Parallel I/O Controller A	
3	PIOB	Parallel I/O Controller B	
4	PIOC	Parallel I/O Controller C	
5	ADC	Analog to Digital Converter	
6	US0	USART 0	
7	US1	USART 1	
8	US2	USART 2	
9	MCI	Multimedia Card Interface	
10	UDP	USB Device Port	
11	TWI	Two-Wire Interface	
12	SPI0	Serial Peripheral Interface 0	
13	SPI1	Serial Peripheral Interface 1	
14	SSC	Synchronous Serial Controller	
15-16	-	Reserved	
17	TC0	Timer/Counter 0	
18	TC1	Timer/Counter 1	
19	TC2	Timer/Counter 2	
20	UHP	USB Host Port	
21	EMAC	Ethernet MAC	
22	ISI	Image Sensor Interface	
23	US3	USART 3	
24	US4	USART 4	
25	US5	USART 5	
26	TC3	Timer/Counter 3	
27	TC4	Timer/Counter 4	
28	TC5	Timer/Counter 5	
29	AIC	Advanced Interrupt Controller	IRQ0
30	AIC	Advanced Interrupt Controller	IRQ1
31	AIC	Advanced Interrupt Controller	IRQ2

Table B.1. Peripheral Identifiers

Appendix C. Address Map (Physical Address Space)

After the execution of the remap command the 4 GB physical address space is separated as shown in the following table. Accessing these addresses directly is only possible if the MMU (memory management unit) is deactivated. As soon as the MMU is activated the visible address space is changed completely. If absolute memory addresses should be accessed within an application, the corresponding address space has first to be mapped to the virtual address space using mmap or ioremap under Linux.

Address (Hex)	Mnemonic	Function
00 0000	Boot Memory	NCS0 or internal ROM or internal SRAM (depending on BMS and REMAP)
10 0000	ROM	Internal ROM 32 kByte
20 0000	SRAM0	Internal SRAM 16 kByte
30 0000	SRAM1	Internal SRAM 16 kByte
50 0000	UHP	USB Host Port
1000 0000	EBI NCS0	Chip Select 0
2000 0000	EBI NCS1	Chip Select 1: SDRAM
3000 0000	EBI NCS2	Chip Select 2
4000 0000	EBI NCS3	Chip Select 3: NAND
5000 0000	EBI NCS4	Chip Select 4
6000 0000	EBI NCS5	Chip Select 5
7000 0000	EBI NCS6	Chip Select 6
8000 0000	EBI NCS7	Chip Select 7
FFFA 0000	TC0, TC1, TC2	3 Timer Counter, 16-Bit
FFFA 4000	UDP	USB Device Port
FFFA 8000	MCI	Multimedia Card / SD-Card Interface
FFFA C000	TWI	Two Wire Interface (I^{2}C)
FFFB 0000	USART0	Synchronous or Asynchronous Serial Port #0
FFFB 4000	USART1	Synchronous or Asynchronous Serial Port #1
FFFB 8000	USART2	Synchronous or Asynchronous Serial Port #2
FFFB C000	SSC	Serial Synchronous Controller (I^{2}S)
FFFC 0000	ISI	Image Sensor Interface
FFFC 4000	EMAC	Ethernet Controller
FFFC 8000	SPI0	Serial Peripheral Interface #0
FFFC C000	SPI1	Serial Peripheral Interface #1
FFFD 0000	USART3	Synchronous or Asynchronous Serial Port #3
FFFD 4000	USART4	Synchronous or Asynchronous Serial Port #4
FFFD 8000	USART5	Synchronous or Asynchronous Serial Port #5
FFFD C000	TC3, TC4, TC5	3 Timer Counter, 16-Bit
FFFE 0000	ADC	Analog Digital Converter
FFFF E800	ECC	Error Correction Controller

Address (Hex)	Mnemonic	Function
FFFF EA00	SDRAMC	SDRAM Controller
FFFF EC00	SMC	Static Memory Controller
FFFF EE00	MATRIX	Bus Matrix User Interface
FFFF F000	AIC	Advanced Interrupt Controller
FFFF F200	DBGU	Debug Unit, including UART
FFFF F400	PIOA	32 Bit Parallel I/O Controller A
FFFF F600	PIOB	32 Bit Parallel I/O Controller B
FFFF F800	PIOC	32 Bit Parallel I/O Controller C
FFFF FC00	PMC	Power Management Controller
FFFF FD00	RSTC	Reset Controller, Battery Powered
FFFF FD10	SHDWC	Shutdown Controller, Battery Powered
FFFF FD20	RTT	Real-time Timer 32 Bit, Battery Powered
FFFF FD30	PIT	Periodic Interval Timer 32 Bit
FFFF FD40	WDT	Watchdog Timer
FFFF FD50	GPBR	4 General Purpose Backup Registers

Table C.1. Physical Address Space

Appendix D. PortuxG20 Pin Assignment

	Processor Pin	Peripheral A	Peripheral B
1	PB31	PCK1	ISI_MCK
2	PB13	RXD5	ISI_D11
3	PB03	SPI1_NPCS0	TIOA5
4	PB01	SPI1_MOSI	TIOB3
5	GND		
6	PA07	**MCCDA**	
7	PA09	**MCDA1**	
8	PA11	**MCDA3**	ETX3
9	PB21	RF0	ISI_D1
10	PA30	SCK2	RXD4
11	PA03	SPI0_NPCS0	MCDB3
12	PA01	SPI0_MOSI	MCCDB
13	GND		
14	**BPC05**	SPI1_NPCS1	A24
15	PB11	RXD3	ISI_D09
16	**BPC10**	CTS3	A25
17	BPC06	CFCE1	TIOB2
18	BPC04	SPI1_NPCS2	A23
19	**HDPB**		
20	PB17	TF0	TCLK4
21	PB07	RXD1	TCLK2
22	PB19	RD0	TIOB5
23	PB29	CTS1	ISI_VSYNC
24	GND		
25	PC01	PCK0	AD1
26	PC03	SPI1_NPCS3	AD3
27	PA23	TWD	ETX2
28	PB22	DSR0	ISI_D2
29	PB05	RXD0	
30	PB24	DTR0	ISI_D4
31	PB27	CTS0	ISI_D7
32	GND		

Table D.1. Pin Assignment PXB Row A

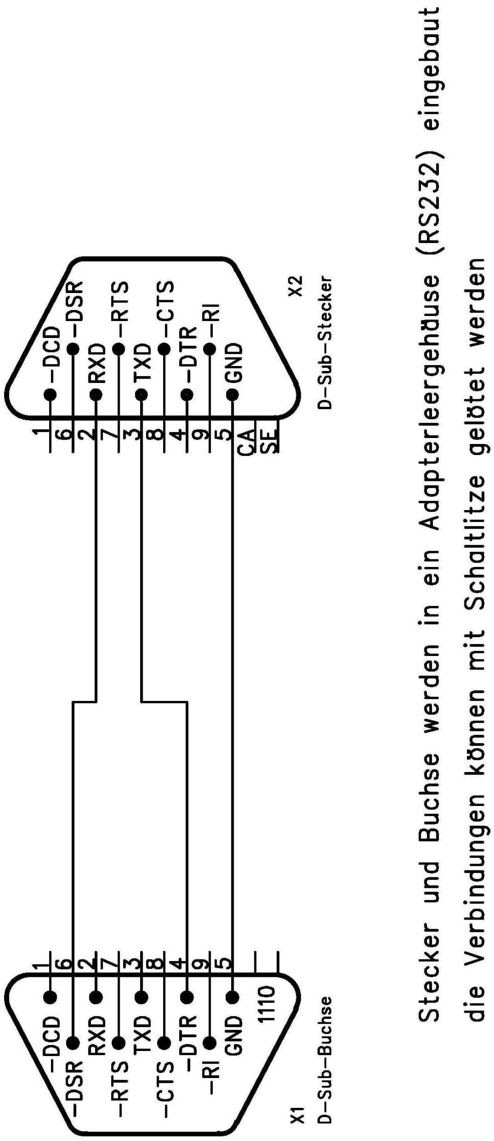
	Processor Pin	Peripheral A	Peripheral B
1	VCC3.3		
2	PB30	PCK0	ISI_HSYNC
3	PB12	TXD5	ISI_D10
4	PB02	SPI1_SPCK	TIOA4
5	PB00	SPI1_MISO	TIOA3
6	PA06	**MCCDA0**	
7	PA08	**MCCK**	
8	PA10	**MCDA2**	ETX2
9	VCC3.3		
10	PB20	RK0	ISID0
11	PA31	SCK0	TXD4
12	PA02	SPI0_SPCK	
13	PA00	SPI0_MISO	MCCDB0
14	VBAT		
15	VCC5.0		
16	PB10	TXD3	ISI_D8
17	BPC08	BNCS4	BRTS3
18	NRST		
19	**HDMB**		
20	VCC3.3		
21	PB16	TK0	TCLK3
22	PB06	TXD1	TCLK1
23	PB28	RTS1	ISI_PCK
24	PB18	TD0	TIOB4
25	PC00	SCK1	AD0
26	PC02	PCK1	AD2
27	PA24	TWCK	ETX3
28	VCC3.3		
29	PB25	RI0	ISI_D5
30	PB04	TXD0	
31	PB26	RTS0	ISI_D6
32	PB23	DCD0	ISI_D3

Table D.2. Pin Assignment PXB Row B

	Processor Pin	Peripheral A	Peripheral B
1	**BD00**		
2	**BD01**		
3	**BD02**		
4	**BD03**		
5	**BD04**		
6	**BD05**		
7	**BD06**		
8	**BD07**		
9	**BD08**		
10	**BD09**		
11	**BD10**		
12	**BD11**		
13	**BD12**		
14	**BD13**		
15	**BD14**		
16	**BD15**		
17	BNRD	BCFOE	
18	BNWR0	BCFWE	BNWE
19	**BNWR1**	BNBS1	BCFIOR
20	BPC15	BIRQ1	BNWAIT
21	BNCS0		
22	BPC11	BNCS2	BSPI0_NPCS1
23	**BA00**	BNBS0	
24	**BA01**	BNBS2	BNWR2
25	**BA02**		
26	**BA03**		
27	**BA04**		
28	**BA05**		
29	**BA06**		
30	**BA07**		
31	**BA08**		
32	**BA09**		

Table D.3. Pin Assignment PXB Row C

Appendix E. Schematics DBGU Adaptor



Stecker und Buchse werden in ein Adapterleergehäuse (RS232) eingebaut
die Verbindungen können mit Schaltlitze gelötet werden

Figure E.1. Schematics DBGU Adaptor

Appendix F. PortuxG20 Electrical Characteristics

Ambient temperature 25℃, unless otherwise indicated

Symbol	Description	Parameter	Min.	Typ.	Max	Unit
V_{CC}	Operating Voltage		3.0	3.3	3.6	V
V_{MEM}	Memory Bus Voltage		1.65	1.8	1.95	V
V_{RES}	Reset Treshhold			2.9		V
T_{RES}	Duration of Reset Pulse		150		280	ms
V_{IH}	High-Level Input Voltage	3.3V	2.0		V_{CC} + 0.3	V
		(PIOC4 - PIOC31) 1.8V	1.26		3.3	V
V_{IL}	Low-Level Input Voltage	3.3V	-0.3		0.8	V
		(PIOC4 - PIOC31) 1.8V	-0.3		0.54	V
P	Normal Operation			540		mW
	Full Load	max.		630		mW
	Stand-By			225		mW
	Power-Down			140		mW
	Full Load with Ethernet			830		mW
V_{BATT}	Battery Voltage		2.0	3.0	V_{CC}	V
I_{BATT}	Battery Current	Ambient temp. = 25℃		5		µA
		Ambient temp. = 70℃			17	µA
		Ambient temp. = 85℃			22	µA

Table F.1. Electrical Characteristics

Appendix G. PortuxG20 Clock Characteristics

Symbol	Description	Dependency	Tolerance	Typical Value	Unit
MAINCK	Main Oscillator frequency			18.432	MHz
SLCK	Slow Clock			32.768	KHz
PLLACK	PLLA Clock	MAINCK		792.000	MHz
PCK	Processor Clock	PLLACK		396.000	MHz
MCK	Master Clock	PCK		132.000	MHz
SDCK	SDRAM Clock	MCK		132.000	MHz
BCK	Baudrate Clock	MCK	1.5%	8.25(max)	MHz
PLLBCK	PLLB Clock	MAINCK		96.000	MHz
USBCK	USB Clock	PLLBCK	0.25%	48.000	MHz

Table G.1. Clock Characteristics

Appendix H. PortuxG20 Environmental Ratings

Symbol	**Description**	**Parameter**	**Operating**		**Storage**		**Unit**
			Min.	**Max.**	**Min.**	**Max.**	
T_A	Ambient temperature		-30	85	-45	85	℃
	Relative Humidity	no condensation		90		90	%RH
	Absolute Humidity		<= Humidity@T_A = 60℃, 90%RH				
	Corrosive Gas		not admissible				

Table H.1. Environmental Ratings

Appendix I. PortuxG20 Dimensions

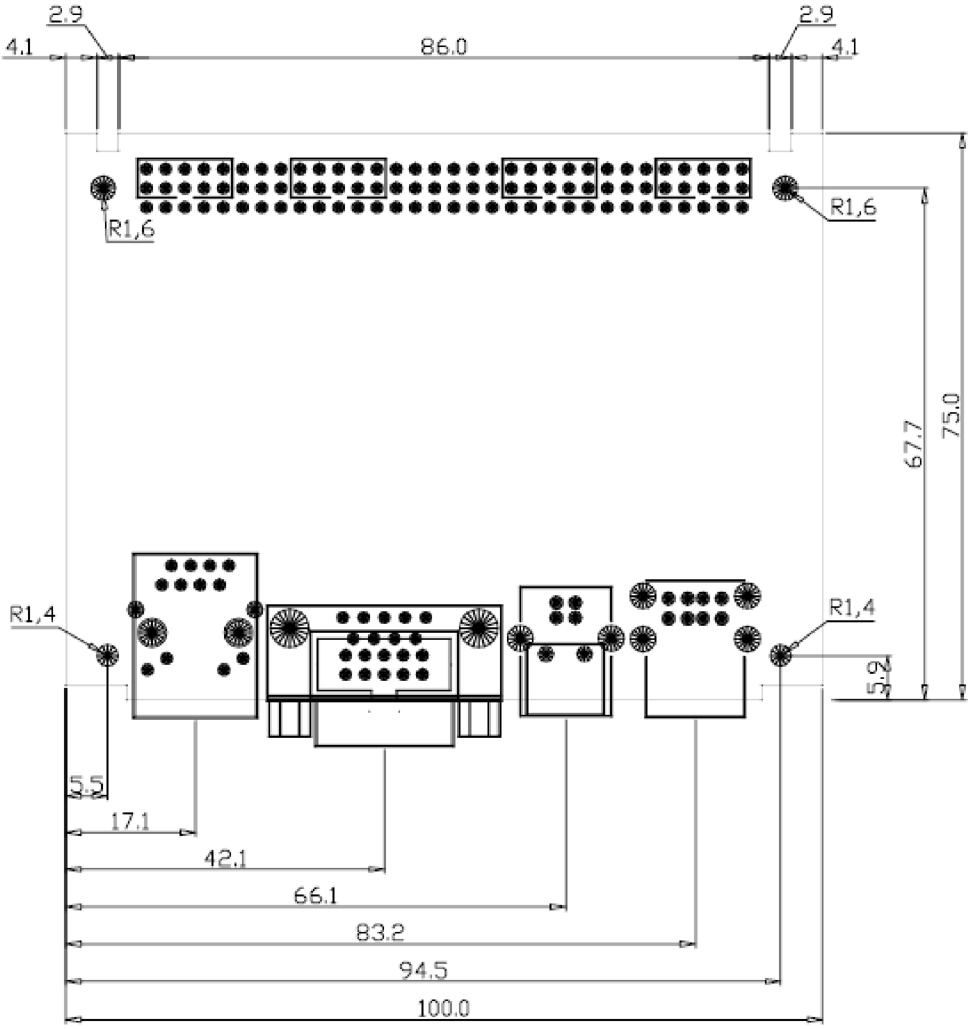


Figure I.1. PortuxG20 Dimensions